

# IT401 – Cloud System Management

## Project Report

**Name:** Renish R, Nandipati Vasanth Kumar Reddy, Pathlavath Sai Kiran Kumar.

**ID:** 202252333, 202252328, 202252330.

**Repository Link:** <https://github.com/renishdotdev/it401-proj>

### Files:

- **Dockerfile:** Container configuration file.
- **scripts/backup.sh:** Bash script to take backup chosen data to ~/backups directory.
- **scripts/rm-cache.sh:** script to remove ~/.cache directory.
- **scripts/rm-logs.sh:** script to remove ~/.logs directory.
- **crontab:** Config file of cron, where commands can be scheduled based on time.
- **entrypoint.sh:** Script which runs during the startup of the container.

### Demonstration:

I am using a raspberry pi, running on pi-OS based on Debian Trixie connected via SSH for this demonstration.

1. Started with cloning my repository and building the docker image.

```
pi@pi5:~ $ git clone https://github.com/renishdotdev/it401-proj
Cloning into 'it401-proj'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 2), reused 12 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 4.28 KiB | 4.28 MiB/s, done.
Resolving deltas: 100% (2/2), done.
pi@pi5:~ $ cd it401-proj/
pi@pi5:~/it401-proj $ ls
crontab Dockerfile entrypoint.sh README.md scripts
pi@pi5:~/it401-proj $ docker build -t cron:latest .
[+] Building 3.2s (16/16) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                         0.1s
=> => transferring dockerfile: 808B                                         0.0s
=> [internal] load metadata for docker.io/library/ubuntu:22.04             2.4s
=> [internal] load .dockerrcignore                                         0.1s
=> => transferring context: 2B                                           0.0s
=> [ 1/11] FROM docker.io/library/ubuntu:22.04@sha256:104ae83764a5119017b8e8d6218fa0832b09df65 0.1s
=> => resolve docker.io/library/ubuntu:22.04@sha256:104ae83764a5119017b8e8d6218fa0832b09df65aa 0.0s
=> [internal] load build context                                         0.1s
=> => transferring context: 6.56kB                                       0.0s
=> CACHED [ 2/11] RUN apt-get update && apt-get install -y      cron      rsync      tzdata 0.0s
=> CACHED [ 3/11] RUN mkdir -p /backups /var/log/cron                0.0s
=> CACHED [ 4/11] COPY scripts/ /usr/local/bin/                      0.0s
=> CACHED [ 5/11] RUN chmod +x /usr/local/bin/*.sh                  0.0s
=> CACHED [ 6/11] COPY crontab /etc/cron.d/backup-cron              0.0s
=> CACHED [ 7/11] RUN chmod 0644 /etc/cron.d/backup-cron            0.0s
=> CACHED [ 8/11] RUN crontab /etc/cron.d/backup-cron               0.0s
=> CACHED [ 9/11] COPY entrypoint.sh /entrypoint.sh                 0.0s
=> CACHED [10/11] RUN chmod +x /entrypoint.sh                        0.0s
=> CACHED [11/11] RUN touch /var/log/cron/cron.log                 0.0s
=> exporting to image                                                 0.2s
=> => exporting layers                                              0.0s
=> => exporting manifest sha256:6cb27850ebf20400e37572339b82b227858f5f973deb3bed53afa0c0fe2dfd 0.0s
=> => exporting config sha256:eb2ccf3d77a79387574411597be4c12e568e29b83a982df076137d4fe97788f8 0.0s
=> => exporting attestation manifest sha256:83dc63b2d6872c6e612978ad3913c339e759e838bba91faef5 0.1s
=> => exporting manifest list sha256:f106b85fce9589817fa4a960918187a3412425d953e44bbfd692c0d24 0.0s
=> => naming to docker.io/library/cron:latest                         0.0s
=> => unpacking to docker.io/library/cron:latest                      0.0s
pi@pi5:~/it401-proj $ |
```

2. Then, I started the ran the container using the command below and ensured its running using “**docker ps**” command.

```
docker run -d --name cron (container name is cron)
--restart unless-stopped (auto restart the container on boot unless stopped by the user)
-e TZ=Asia/Kolkata (configuring time zone)
-v /home:/host/home:ro (mounting /home directory of host to /host/home in docker volume)
-v ~/backups:/backups (mounting ~/backups of host to /backups in docker volume)
-v cron-logs:/var/log/cron
cron:latest (choosing the built image)
```

```
pi@pi5:~/it401-proj $ docker run -d \
--name cron \
--restart unless-stopped \
-e TZ=Asia/Kolkata \
-v /home:/host/home:ro \
-v ~/backups:/backups \
-v cron-logs:/var/log/cron \
cron:latest
a732232730b7046c1172ce2b1c3a552040e0ab8257d9a6e11d31a87046744aca
pi@pi5:~/it401-proj $ docker ps
CONTAINER ID   IMAGE          COMMAND           CREATED          STATUS          PORTS
a732232730b7   cron:latest   "/entrypoint.sh"   9 seconds ago   Up 8 seconds
                  cron
9b28a8b769fa   docker.n8n.io/n8nio/n8n:latest "tini -- /docker-ent..."   9 days ago     Up 5 hours      0.0.0.0:5678->5678/tcp, [::]:5678->5678/tcp  n8n
pi@pi5:~/it401-proj $ |
```

3. In this repo, I have configured to run the scripts every minute, so I left it to run for few minutes and checked the ~/backups directory.

```
pi@pi5:~/it401-proj $ ls -lh ~/backups/
total 288K
drwxr-xr-x 4 root root 4.0K Nov 24 16:00 20251124_103001
drwxr-xr-x 4 root root 4.0K Nov 24 16:01 20251124_103101
drwxr-xr-x 4 root root 4.0K Nov 24 16:02 20251124_103201
drwxr-xr-x 4 root root 4.0K Nov 24 16:03 20251124_103301
drwxr-xr-x 4 root root 4.0K Nov 24 16:04 20251124_103401
drwxr-xr-x 4 root root 4.0K Nov 24 16:05 20251124_103501
drwxr-xr-x 4 root root 4.0K Nov 24 16:06 20251124_103601
drwxr-xr-x 4 root root 4.0K Nov 24 16:07 20251124_103701
drwxr-xr-x 4 root root 4.0K Nov 24 16:08 20251124_103801
drwxr-xr-x 4 root root 4.0K Nov 24 16:09 20251124_103901
drwxr-xr-x 4 root root 4.0K Nov 24 16:10 20251124_104001
drwxr-xr-x 4 root root 4.0K Nov 24 16:11 20251124_104101
drwxr-xr-x 4 root root 4.0K Nov 24 16:12 20251124_104201
drwxr-xr-x 4 root root 4.0K Nov 24 16:13 20251124_104301
drwxr-xr-x 4 root root 4.0K Nov 24 16:14 20251124_104401
drwxr-xr-x 4 root root 4.0K Nov 24 16:15 20251124_104501
drwxr-xr-x 4 root root 4.0K Nov 24 16:16 20251124_104601
drwxr-xr-x 4 root root 4.0K Nov 24 16:17 20251124_104701
drwxr-xr-x 4 root root 4.0K Nov 24 16:18 20251124_104801
drwxr-xr-x 4 root root 4.0K Nov 24 16:19 20251124_104901
drwxr-xr-x 4 root root 4.0K Nov 24 16:20 20251124_105001
```

Each backup taken is stored here with the timestamp as the directory name. Each contain the data stored in ~/.config and ~/Documents directory of each user. My raspberry pi only has 1 user.

```

pi@pi5:~/backups/20251124_103001 $ ls
configs documents
pi@pi5:~/backups/20251124_103001 $ cd configs/
pi@pi5:~/backups/20251124_103001/configs $ ls
pi
pi@pi5:~/backups/20251124_103001/configs $ cd pi
pi@pi5:~/backups/20251124_103001/configs/pi $ ls
bluedevilglobalrc          kded5rc           PlasmaDiscoverUpdates
breezerc                  kdedefaults        plasma-localerc
chromium                  kdeglobals         plasma-org.kde.plasma.desktop-appletsrc
Code                      kde.org            plasma-shellrc
dconf                     kglobalshortcutsrc   plasma-welcomerc
dolphinrc                 konsolerc          powermanagementprofilessrc
drkonqirc                 konsoleshconfig    procps
fontconfig                ksmserverrc       pulse
calculator                ksplashrc         qt5ct
geany                     ktimezonedrc     qt6ct
gtk-3.0                   kwalletrc          'Raspberry Pi'
gtk-4.0                   kwinoutputconfig.json  rpcc
gtkrc                     kwinrc             sddmthemeinstallerrc
gtkrc-2.0                 labwc              session
kactivitymanagerdrc       libaccounts-glib   Trolltech.conf
kactivitymanagerd-statsrc libreoffice        user-dirs.dirs
kanshi                    lxpanel-pi         user-dirs.locale
kconf_updaterc            lxsession          vlc
KDE                       lxterminal        wf-panel-pi
kdeconnect                openbox            xsettingsd
pi@pi5:~/backups/20251124_103001/configs/pi $ |

```

#### 4. Checking script logs using “**docker logs**” command.

```

pi@pi5:~ $ docker logs cron
Starting cron container...
Cron jobs configured:
# Cron schedule format: minute hour day month weekday command
#
# Run backup daily at 2 AM
# 0 2 * * * /usr/local/bin/backup.sh >> /var/log/cron/cron.log 2>&1

# Run cache cleanup daily at 3 AM
# 0 3 * * * /usr/local/bin/rm-cache.sh >> /var/log/cron/cron.log 2>&1

# Run logs cleanup weekly on Sunday at 4 AM
# 0 4 * * 0 /usr/local/bin/rm-logs.sh >> /var/log/cron/cron.log 2>&1

# For testing: Run all scripts every 5 minutes (comment out for production)
* * * * * /usr/local/bin/backup.sh >> /var/log/cron/cron.log 2>&1
* * * * * /usr/local/bin/rm-cache.sh >> /var/log/cron/cron.log 2>&1
* * * * * /usr/local/bin/rm-logs.sh >> /var/log/cron/cron.log 2>&1

# Empty line required at end of crontab
Starting cron daemon...
Container ready. Tailing logs...
== Logs cleanup finished at Mon Nov 24 11:52:01 UTC 2025 ===
Backing up config for user: pi
Cleaning cache for user: pi
  Freed: 0B
Total cache freed: 0B
== Cache cleanup finished at Mon Nov 24 11:52:01 UTC 2025 ===
Backing up documents for user: pi
Backup completed: 1 configs, 1 documents backed up
Backup location: /backups/20251124_115201 (Size: 146M)
== Backup finished at Mon Nov 24 11:52:01 UTC 2025 ===
== Logs cleanup started at Mon Nov 24 11:53:01 UTC 2025 ===
== Cache cleanup started at Mon Nov 24 11:53:01 UTC 2025 ===
== Backup started at Mon Nov 24 11:53:01 UTC 2025 ===
Total logs freed: 0B
== Logs cleanup finished at Mon Nov 24 11:53:01 UTC 2025 ===
Backing up config for user: pi
Cleaning cache for user: pi
  Freed: 0B
Total cache freed: 0B

```